

# Rapport de TIPE

Timothée Darcet

15 juin 2017

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Corps principal</b>	<b>3</b>
2.1	La méthode de l'enveloppe visuelle . . . . .	3
2.2	Une formalisation mathématique . . . . .	4
2.3	L'implémentation en machine . . . . .	5
2.4	Les tests effectués . . . . .	7
<b>3</b>	<b>Conclusion</b>	<b>7</b>
<b>A</b>	<b>Annexe</b>	<b>7</b>
<b>B</b>	<b>Annexe</b>	<b>8</b>
<b>C</b>	<b>Annexe</b>	<b>11</b>

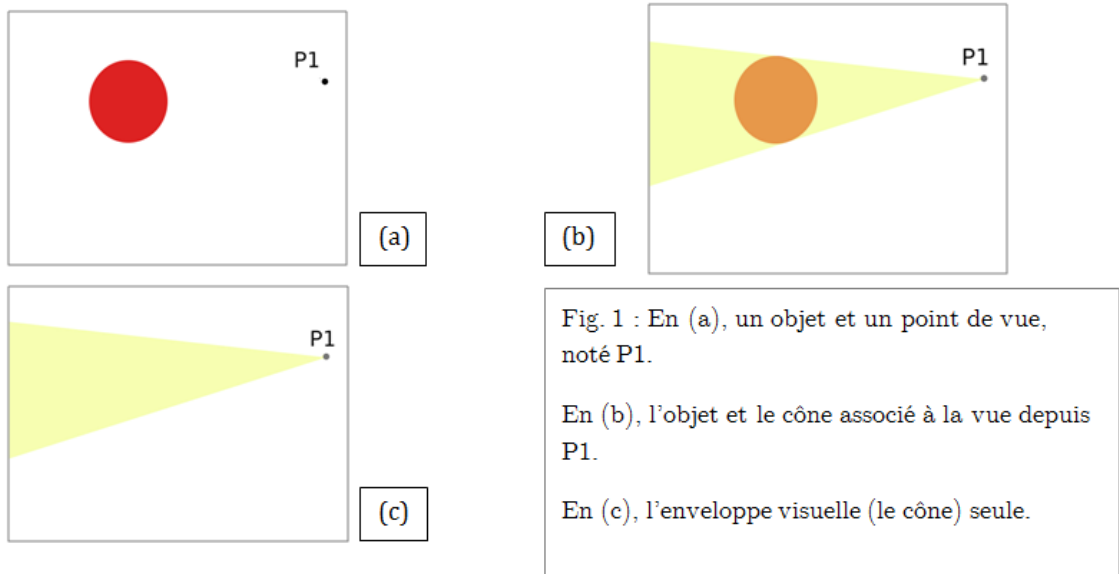
# 1 Introduction

Pour optimiser le processus de reconstitution d'objets en trois dimensions, nous nous étions proposé de choisir et d'implémenter en Python un tel algorithme de reconstitution, et d'en étudier l'efficacité. C'est ce que nous avons fait, en choisissant l'algorithme dit d'« enveloppe visuelle » étudié notamment par B. Baumgart [1] et A. Laurentini [2]. Il présente l'avantage d'être conceptuellement assez simple, et d'être implémentable de manière relativement légère. Nous avons donc réalisé une implémentation de cet algorithme, puis avons effectué des tests pour en étudier l'efficacité, et pouvoir analyser la pertinence des choix réalisés durant la construction de l'algorithme. Je me suis personnellement occupé de la partie de modélisation en 3D du projet, tandis que mon binôme s'est occupé du prétraitement des photographies pour les rendre exploitables. Pour s'assurer du fait que le modèle produit par l'algorithme n'est, dans des cas idéalisés, pas trop éloigné de l'objet original, j'ai aussi démontré des résultats théoriques portant l'algorithme.

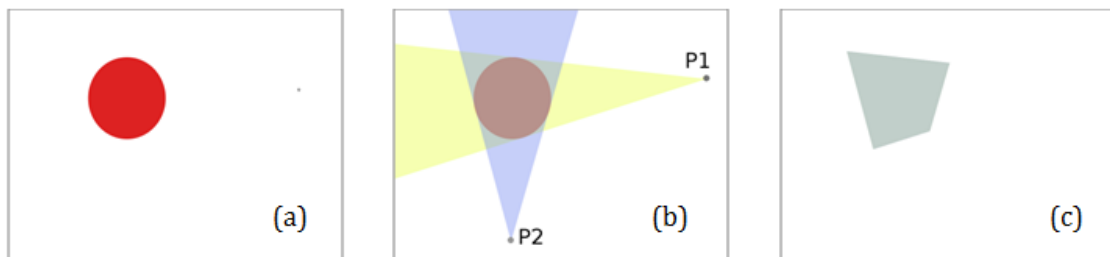
## 2 Corps principal

### 2.1 La méthode de l'enveloppe visuelle

L'« enveloppe visuelle » d'un objet A est l'objet maximal qui donne la même silhouette que A quand on l'observe depuis un ensemble de points de vue donné. Ainsi, l'enveloppe visuelle de A depuis un point de vue unique est un cône dont le sommet est placé sur ce point de vue, et qui a la forme de la silhouette de A depuis ce point de vue (fig. 1).



Pour obtenir l'enveloppe visuelle d'un objet depuis plusieurs points de vue, on peut alors simplement considérer l'intersection des enveloppes visuelles obtenues depuis chacun des points de vue. Un exemple (en deux dimensions, pour simplifier) est donné en fig. 2.



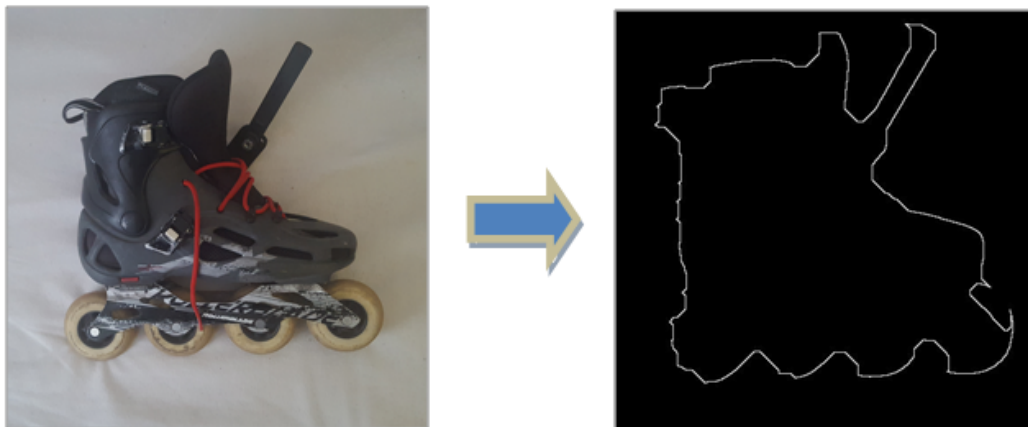
En (a), un objet.

En (b), les cônes correspondant aux deux points de vue P1 et P2.

En (c), l'intersection des deux cônes, soit l'enveloppe

En multipliant les points de vue, l'enveloppe visuelle se rapproche de l'objet original. On peut donc utiliser cette méthode pour reconstituer le modèle 3D d'un objet, simplement à partir de photographies. Se dégagent alors deux étapes majeures pour cette reconstitution :

- Le traitement de la photographie, pour dégager la silhouette de l'objet :



- La reconstitution à proprement parler, qui consiste à créer les cônes correspondant aux silhouettes, puis à en calculer l'intersection.

Ma contribution personnelle fut une théorisation de l'enveloppe visuelle, qui permet d'établir des résultats dessus, et la construction d'un système de représentation en machine d'objets en 3D, ainsi que de tous les algorithmes l'accompagnant, en particulier les algorithmes d'intersection, de constructions de cônes, et de simulation de la prise d'une photographie. Cette construction fut réalisée en langage python.

## 2.2 Une formalisation mathématique

Formellement, une photographie est une projection plane relative à un foyer. Ainsi, si on munit  $\mathbb{R}^3$  de son produit scalaire canonique  $(\cdot|\cdot)$ , on peut, pour un foyer  $f \in \mathbb{R}^3$  et un point du plan image  $p$ , poser

$$p_{f,p} : \mathbb{R}^3 \setminus \{x \in \mathbb{R}^3 | (x-f|p-f) = 0\} \rightarrow \mathbb{R}^3$$

$$x \mapsto f + \frac{\|p-f\|^2}{(x-f|p-f)} \cdot (x-f)$$

On remarque que pour tout  $x \in \mathbb{R}^3$ , il existe  $\lambda \in \mathbb{R}$  tel que  $p(x) - f = \lambda(x - f)$ , donc  $f$ ,  $x$  et  $p(x)$  sont alignés.

De plus pour tout  $x$  pour lequel  $p(x)$  est défini,  $(p(x) - f|p - f) = (p - f|p - f)$  donc  $p(x)$  est bien dans le plan de vecteur normal  $p - f$  qui passe par  $p$ .

Ainsi  $p$  correspond bien à une photographie.

On peut alors poser, pour une partie  $A$  de  $\mathbb{R}^3$  et un ensemble  $V$  de couples  $(f, p) \in \mathbb{R}^3$  où  $f$

représente un foyer et  $p$  un point d'un plan image, l'enveloppe visuelle de  $A$  relative à l'ensemble de points de vues  $V$  :

$$ev(V, A) = \bigcap_{(f,p) \in V} \{f + \lambda \cdot (y - f), \lambda \in \mathbb{R}_+, y \in p_{f,p}(A)\}$$

(On notera que ceci correspond au cas théorique où le plan de projection est situé entre le foyer et l'objet, alors qu'une photographie réelle utilise un plan de projection tel que le foyer soit entre ce plan et l'objet. Cela ne perturbera pas les résultats.)

L'enveloppe visuelle étant formalisée, on peut montrer des résultats dessus.

Premièrement, si un objet est bien entièrement du "bon" côté des plans images choisis, c'est-à-dire que chaque plan image est entre l'objet et le foyer correspondant, alors son enveloppe visuelle ne dépend pas des plans images choisis, et elle peut s'exprimer plus simplement :

$$ev(V, A) = \bigcap_{(f,p) \in V} \{f + \lambda \cdot (x - f), \lambda \in \mathbb{R}_+, x \in A\}$$

La démonstration pourra être retrouvée annexe A.

Un corollaire est que pour un objet qui satisfait ces hypothèses, l'objet sera toujours inclus dans son enveloppe visuelle relative à un ensemble de points de vues quelconques.

Un autre résultat que l'on peut alors montrer est que, étant donné une sphère  $S$  qui contient l'objet  $A$ , l'enveloppe visuelle de  $A$  relative aux points de vues situés sur  $S$  est incluse dans l'enveloppe convexe de  $A$ . La démonstration de ce résultat est fournie en annexe B.

## 2.3 L'implémentation en machine

Deux principaux types de stockage en mémoire des objets 3D existent :

- L'approximation par des polyèdres.
- La modélisation par des "voxels", c'est-à-dire des "pixels de volume". L'espace est alors représenté par une grille finie, où chaque case peut prendre plusieurs états, par exemple "remplie" ou "vide".

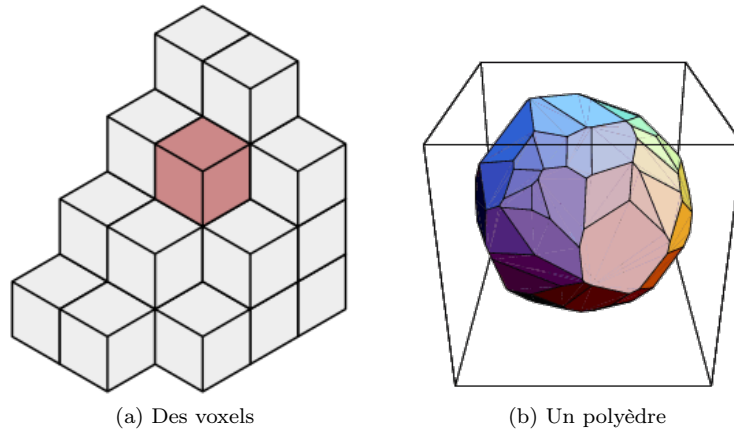


FIGURE 1 – Une comparaison entre polyèdre et voxels

La modélisation précise que nous avons choisi est la modélisation par "polyèdre aux arêtes ailées" ("winged edge polyhedron") introduite par B. Baumgart. Elle utilise quatre types d'objets distincts :

- Les objets polyèdres, qui stockent une liste de faces, une liste de sommets et une liste d'arêtes
- Les sommets, qui stockent leurs trois coordonnées ainsi qu'un lien vers une de leurs arêtes
- Les faces, qui stockent un lien vers une arête adjacente
- Les arêtes, qui stockent huit liens, nommés selon la figure 2 :
  - PVT et NVT pointent vers les sommets aux extrémités de l'arête
  - NFACE et PFACE pointent vers les faces adjacentes à l'arête

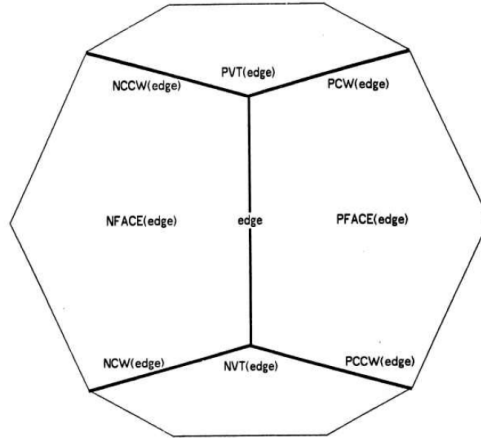


FIGURE 2 – Les liens que stockent une arête dans le modèle du "winged edge polyhedron"

- PCW, NCW, PCCW et NCCW pointent vers les quatre "ailes" (qui donnent ce nom à la modélisation) : ce sont les arêtes qui ont un sommet et une face en commun avec l'arête considérée.

Ainsi un polyèdre cohérent peut être défini. On remarquera que seuls les sommets possèdent des informations sur les positions des éléments dans l'espace, tandis que les arêtes stockent toutes les informations de structure (quels éléments sont liés auxquels). Les faces, quant à elles, donnent des indication sur le caractère visuel du polyèdre : ce sont elles qui vont bloquer ou non la lumière, et ainsi elles seront utiles pour simuler une vue de l'objet.

Cette dissociation des informations stockées en mémoire est intéressante : ainsi, on peut par exemple facilement déformer un polyèdre tout en gardant sa topologie, en appliquant une fonction vectorielle aux seules coordonnées des sommets.

On ne détaillera pas toutes les fonctions utilisées pour la modélisation, mais on se penchera tout de même sur l'algorithme d'intersection de polyèdres, qui est le plus complexe. Dans l'implémentation que nous avons réalisé, nous définissons deux nouveaux types d'objets : les objets **polyIntersection**, qui stockent toutes les informations relative à une intersection en cours de deux polyèdres, et les objets **intersectors**, qui stockent les informations relatives à l'intersection d'une arête d'un polyèdre avec une face de l'autre polyèdre. Ainsi les

- On inventorie tous les cas d'intersections face-arête qui existent, et on les stocke dans une liste d'intersecteurs
- On construit les arêtes du polyèdre résultat
- On construit les faces du polyèdre résultat

**Le test d'intersection arête-face** Le test d'intersection arête/face se déroule en deux étapes :

- On teste si le segment intersecte le plan de la face considérée.
- Si c'est le cas, on teste si l'intersection est incluse dans la face considérée.

Pour tester si une arête intersecte un plan, il suffit de tester si les deux extrémités de l'arête sont du même côté du plan ou non. Pour cela, on calcule le vecteur normal au plan  $n$ . Si  $p$  est un point du plan et  $a$  et  $b$  sont deux points, on a que  $a$  et  $b$  sont du même côté du plan si et seulement si  $(a - p|n)$  et  $(b - p|n)$  sont de même signe, soit si et seulement si  $(a - p|n) \cdot (b - p|n) \geq 0$ . Dans notre cas, pour calculer  $n$  on peut effectuer le produit vectoriel de deux arêtes successives, et pour  $p$  on peut prendre un point quelconque de la face.

Pour tester si le point d'intersection calculé est inclus dans la face, on va calculer les produits vectoriels de la forme  $v_n = (a_n - i) \wedge (a_{n+1} - i)$ , où les  $a_n$  sont les sommets de la face, et  $i$  est le point d'intersection entre l'arête et le plan de la face. Nous admettrons que, pour une face convexe,  $i$  est intérieur à la face si et seulement si  $(v_n|v_{n+1})$  deviendra négatif pour un certain  $n$ .

**La construction des arêtes** Pour reconstruire les arêtes, on en distingue deux types :

**Les arêtes intérieures** sont des anciennes arêtes d'un des deux polyèdres originaux (ou des morceaux de ceux-ci) qui se trouvaient à l'intérieur de l'autre.

**Les arêtes de surface** sont des nouvelles arêtes, formées par l'intersection de deux faces.

On peut retrouver les arêtes intérieures car elle forment des graphes connectés à des points d'intersections face-arête entre les deux polyèdres. Ainsi, on itère sur les intersecteurs répertoriés plus tôt

## 2.4 Les tests effectués

À partir de cette modélisation, j'ai réalisé des tests sur l'algorithme d'intersection pour obtenir des informations sur le temps d'exécution de cet algorithme pour divers complexités de polyèdres initiaux. Nous avons préféré une analyse du temps d'exécution pratique plutôt qu'une étude de la complexité théorique. En effet, si cette dernière donne des bonnes informations asymptotiques, elle ne donne aucun contrôle sur la vitesse de convergence, et un algorithme en  $O(n^2)$  pourra être en pratique beaucoup plus rapide à exécuter qu'un autre qui est en  $O(n)$ , simplement parce que les  $n$  considérés ne deviennent jamais très grands. De plus, une information sur le temps d'exécution, en secondes, est bien plus facilement interprétable qu'une information de complexité. Ainsi, des tests simples sur le programme que j'ai réalisé montrent qu'une intersection simple, avec deux polyèdre ayant de l'ordre de la dizaine de sommets, prend en moyenne 0,15s sur un ordinateur portable, le processeur y étant cadencé à 2 GHz. Ceci, bien sûr, n'est que le temps d'exécution sur une machine particulière, et en utilisant des machines plus puissantes, ce temps pourrait être considérablement réduit. Mais il faut reprendre les objectifs que nous nous étions fixés : nous cherchions un algorithme efficace, qui renvoie des résultats les plus fidèles possibles tout ayant un temps d'exécution le plus court possible. Ici, les résultats obtenus signifient que pour pouvoir reconstituer un modèle 3D avec une précision acceptable, en un temps raisonnable (c'est-à-dire réaliser de l'ordre de 10 à 100 intersections de polyèdres qui possèdent de l'ordre de 1000 sommets), il faudrait une machine très puissante.

## 3 Conclusion

En conclusion, le modèle construit ici est modérément efficace. Le traitement d'image effectué par mon binôme est, lui aussi, relativement lent (de l'ordre de la minute sur des ordinateurs personnels). Nous obtenons donc un programme qui fonctionne en un temps raisonnable, et dont nous savons qu'il renverra un modèle proche de l'original, mais qui n'est pas aussi efficace que ce qu'on avait espéré.

## A Annexe

Soit une partie  $A$  de  $\mathbb{R}^3$  et un ensemble  $V$  de couples  $(f, p) \in \mathbb{R}^3$  où  $f$  représente un foyer et  $p$  un point d'un plan image. On suppose que pour tout élément  $x$  de  $A$ , pour tout  $(f, p) \in V$ ,

$$(x - f|p - f) > 0$$

Ce qui signifie que le plan normal à  $p - f$  passant par  $p$  (le plan image) est entre  $f$  et  $x$ . Soit  $(f, p) \in V$

$$\begin{aligned} \{f + \lambda \cdot (y - f), \lambda \in \mathbb{R}_+, y \in p_{f,p}(A)\} &= \{f + \lambda \cdot \frac{\|p - f\|^2}{(x - f|p - f)} \cdot (x - f), \lambda \in \mathbb{R}_+, y \in p_{f,p}(A)\} \\ &= \{f + \mu \cdot (x - f), \mu \in \mathbb{R}_+, y \in p_{f,p}(A)\} \end{aligned}$$

On a donc bien que

$$ev(V, A) = \bigcap_{(f,p) \in V} \{f + \lambda \cdot (x - f), \lambda \in \mathbb{R}_+, x \in A\}$$

## B Annexe

Nous nous plaçons dans le  $\mathbb{R}$ -espace vectoriel  $E = \mathbb{R}^3$ , muni de sa structure d'espace euclidien canonique, pour laquelle le produit scalaire sera noté  $(\cdot|\cdot)$

Pour tout  $x \in E$ , on définit le projecteur de foyer  $x$  comme :

$$\begin{aligned} p_x &: E \setminus \{x\} \rightarrow S(x, 1) \\ y &\mapsto x + \frac{1}{\|y-x\|} \cdot (y-x) \end{aligned}$$

(On notera  $S(x, r)$ ,  $B_o(x, r)$  et  $B_f(x, r)$  respectivement la sphère, la boule ouverte et la boule fermée de centre  $x$  et de rayon  $r$ )

On définit la silhouette d'une partie  $A$  de  $E$  vue depuis un point  $x$  comme son image directe par  $p_x$ .

*Note : Cela ne correspond pas exactement à une photographie, qui elle est une projection sur un plan. Toutefois, on admettra que si un objet est entièrement du côté du plan focal opposé au foyer (ce qui est toujours vérifié pour une photographie), le "cône" reconstruit à partir de sa photographie est le même que celui reconstruit à partir de la projection sphérique, et que donc la définition de l'enveloppe visuelle que l'on va donner ici correspond donc bien à la définition "naturelle".*

On définit ensuite le cône de sommet  $x$  et de silhouette  $A \subset S(x, 1)$  comme suit :

$$C(x, A) = \{x + t \cdot (y - x), y \in A, t \in \mathbb{R}_+\}$$

On peut alors définir l'enveloppe visuelle d'une partie  $A$  de  $E$  selon un ensemble de points de vues  $V$  :

$$ev(V, A) = \bigcap_{x \in V} C(x, p_x(A))$$

On notera aussi  $conv(A)$  l'enveloppe convexe de  $A$ , et on admettra que :

$$conv(A) = \left\{ \sum_{i=1}^n \lambda_i x_i \mid n \in \mathbb{N}^*, (x_i)_{1 \leq i \leq n} \in E^n, (\lambda_i)_{1 \leq i \leq n} \in [0, 1]^n \text{ t.q. } \sum_{i=1}^n \lambda_i = 1 \right\}$$

J'ai montré dans un premier temps le lemme 1 :

**Lemme 1.** *Si  $A$  est un fermé de  $E$ , et il existe  $b \in E$  et  $M \in \mathbb{R}$  tels que  $A \subset B_o(b, M)$ ,*

$$\text{Alors } ev(S(b, M), A) \subset B_o(b, M)$$

*Démonstration.* On supposera, quitte à faire effectuer une translation à  $A$ , que  $b = 0$

Supposons qu'il existe  $x \in ev(S(0, M), A)$  tel que  $\|x\| \geq M$

— Si  $\|x\| > M$

On considère  $\beta$  un vecteur orthogonal à  $x$  de norme 1 ( $\beta$  existe car  $E$  est un espace euclidien de dimension 3)

On pose  $y = \frac{M^2}{\|x\|^2} \cdot x + M \sqrt{1 - \frac{M^2}{\|x\|^2}} \cdot \beta$

On remarque que

$$\begin{aligned} \|y\|^2 &= \frac{M^4}{\|x\|^2} + M^2 \left(1 - \frac{M^2}{\|x\|^2}\right) \\ &= M^2 \end{aligned}$$

Donc que  $y \in S(0, M)$

Et que

$$\begin{aligned} (y|x - y) &= (y|x) - \|y\|^2 \\ &= M^2 - M^2 \\ &= 0 \end{aligned}$$



Or  $y \in S(0, M)$  donc  $x \in C(y, p_y(A))$ , donc il existe  $u \in p_y(A)$  et  $t \in \mathbb{R}^+$  tels que  $x = y + t \cdot (u - y)$ .

Comme  $\|x\| > M$ ,  $t \neq 0$

Donc  $u - y = \frac{1}{t} \cdot (x - y)$ , et donc  $(u - y|y) = 0$

De plus  $u \in p_y(A)$  donc il existe  $v \in A$  tel que  $u = y + \frac{1}{\|v-y\|} \cdot (v - y)$

Donc  $v = y + \|v - y\| \cdot (u - y)$

Comme  $(u - y|y) = 0$ ,

$$\begin{aligned}\|v\|^2 &= \|y\|^2 + \|v - y\|^2 \cdot \|u - y\|^2 \\ &\geq M^2\end{aligned}$$

Or, par hypothèse,  $v \in A$  implique  $\|v\| < M$

On a une contradiction.

— Si  $\|x\| = M$

$A$  est fermé et borné car inclus dans  $B_o(0, M)$ , donc  $A$  est compact (On a  $\dim(E) = 3$ ).

Et comme  $u \mapsto \|u\|$  est continue, elle admet un maximum  $n_m$  sur  $A$ . Il existe donc  $x_0 \in A$  tel que  $\|x_0\| = n_m$

$x_0 \in B_o(0, M)$  donc  $n_m < M$

Soit  $y$  un vecteur de  $S(0, M)$  tel que  $\|y - x\| < \frac{M - n_m}{2}$  et  $y \neq x$

$y \in S(0, M)$  donc  $x \in C(y, p_y(A))$

Donc il existe  $v \in A$  et  $t \in \mathbb{R}^+$  tel que  $x = y + \frac{t}{\|v-y\|} \cdot (v - y)$

$x \neq y$  donc  $t \neq 0$  et  $v = y + \frac{\|v-y\|}{t} \cdot (x - y)$

On note

$$\begin{aligned}f &: \mathbb{R} \rightarrow \mathbb{R} \\ \lambda &\mapsto \|y\|^2 + \lambda^2 \cdot \|x - y\|^2 + 2\lambda(y|x - y)\end{aligned}$$

(On a alors  $f(\lambda) = \|y + \lambda \cdot (x - y)\|^2$ , et donc  $\|v\|^2 = f(\frac{\|v-y\|}{t})$ )

Or  $f'(\lambda) = 2\lambda \|x - y\|^2 + 2(y|x - y)$

Donc  $f(\lambda)$  atteint un minimum en

$$\lambda = \frac{(y|y - x)}{\|x - y\|^2} = \frac{\|y\|^2 - (y|x)}{\|x\|^2 + \|y\|^2 - 2(x|y)} = \frac{1}{2}$$

(Car  $\|x\| = \|y\|$ )

Ainsi

$$\begin{aligned}\|v\|^2 &= f\left(\frac{\|v-y\|}{t}\right) \\ &\geq f\left(\frac{1}{2}\right) \\ &\geq \left\|y + \frac{1}{2}(x - y)\right\|^2 \\ &\geq \left(\|y\| - \frac{1}{2}\|x - y\|\right)^2\end{aligned}$$

Donc  $\|v\| \geq \|y\| - \frac{1}{2}\|x - y\|$

Or, par construction de  $y$ ,  $\|y - x\| < \frac{M - n_m}{2}$

Donc  $\|v\| > n_m$

Or  $v \in A$  donc  $\|v\| \leq n_m$  On a encore une contradiction.

Dans les deux cas on a une contradiction. Donc  $\|x\| < M$

□

J'ai ensuite montré le théorème 1 :

**Théorème 1.** *L'enveloppe visuelle d'un fermé  $A$  de  $E$  vu depuis une sphère englobant  $A$  est incluse dans l'enveloppe convexe de  $A$ .*

*En d'autres termes,*

$$\forall A \subset E \text{ t.q. } A \text{ fermé et borné, } \forall (b, M) \in E \times \mathbb{R} \text{ t.q. } A \subset B_o(b, M), \\ \text{On a } ev(S(b, M), A) \subset conv(A)$$

*Démonstration.* Soit  $A$  un fermé de  $E$

On considérera  $A$  non vide (Si  $A$  est vide,  $ev(S(b, M), A) = conv(A) = A = \emptyset$ )

Soit  $a \in ev(S(b, M), A)$

Quitte à effectuer une translation sur  $A$  et  $b$ , on va considérer que  $a = 0_E$ .

— Si  $0 \in A$ , immédiatement  $0 \in conv(A)$

— Si  $0 \notin A$ , on a que  $\forall x \in S(b, M), 0 \in C(x, p_x(A))$

Donc

$$\exists t \in \mathbb{R}_+, \exists y \in p_x(A) \text{ t.q. } 0 = x + t \cdot (y - x)$$

Or  $y \in p_x(A)$  donc il existe  $z$  dans  $A$  tel que  $p_x(z) = y$ , soit

$$y - x = \frac{1}{\|z - x\|} \cdot (z - x)$$

Soit  $x_0 \in S(0, 1)$

On note alors

$$\begin{array}{ccc} d & : & \mathbb{R} \rightarrow E \\ & & t \mapsto t \cdot x_0 \end{array}$$

$d(\mathbb{R})$  est la droite passant par  $x_0$  et par  $0$ .

De plus  $A \subset B_o(b, M)$  et  $0 \in ev(S(b, M), A)$  donc, selon le lemme 1,  $\|0 - b\| < M$

Ainsi  $\|d(0) - b\| < M$ , et comme  $\lim_{t \rightarrow +\infty} \|d(t)\| = +\infty$  et  $t \mapsto \|d(t) - b\|$  est une application continue, selon le théorème des valeurs intermédiaires,

$$\exists t_1 \in \mathbb{R} \setminus \{0\} \text{ tel que } \|d(t_1)\| = M$$

On note  $x_1 = d(t_1)$

Comme  $0 \in C(x_1, p_{x_1}(A))$ , il existe  $v \in A$  et  $\lambda \in \mathbb{R}_+$  tels que

$$0 = x_1 + \lambda(v - x_1)$$

De plus  $\lambda \neq 0$ , donc

$$v = \frac{\lambda - 1}{\lambda} \cdot x_1 = t_1 \cdot \left(1 - \frac{1}{\lambda}\right) x_0$$

Ainsi, selon le signe de  $t_1 \cdot \left(1 - \frac{1}{\lambda}\right)$ , on aura  $p_0(v) = x_0$  ou  $p_0(v) = -x_0$

Cela est vrai pour tout  $x_0$  dans  $S(0, 1)$  donc, en notant  $P = p_0(A)$  et  $P' = \{-a, a \in P\}$ ,  $P \cup P' = S(0, 1)$

On cherche alors à montrer que  $P \cap P' \neq \emptyset$

Comme  $A$  est fermé et borné en dimension finie,  $A$  est compact, et comme  $p_0$  est continue,  $P = p_0(A)$  est compact aussi.

Et comme  $a \mapsto -a$  est continue,  $P'$  est compact aussi.

Soit alors  $u \in P$  et  $v$  un vecteur orthogonal à  $u$  de norme 1

On note

$$\begin{array}{ccc} \gamma & : & [0, 1] \rightarrow S(0, 1) \\ & & t \mapsto \cos(t\pi) \cdot u + \sin(t\pi) \cdot v \end{array}$$

$\gamma$  est continue, donc  $\gamma^{-1}(P)$  et  $\gamma^{-1}(P')$  sont fermés.

$\gamma(0) = u \in P$  donc  $\gamma^{-1}(P) \neq \emptyset$  et on peut donc considérer  $t = \sup \gamma^{-1}(P) = \max \gamma^{-1}(P)$  (Car  $\gamma^{-1}(P)$  est fermé)

— Si  $t = 1$ ,  $-u \in P$ , et comme on a aussi  $u \in P$ ,  $u \in P \cap P'$

— Si  $t < 1$ , il existe  $n_0 \in \mathbb{N}$  tel que  $\forall n \geq n_0, t + \frac{1}{n} \leq 1$  Alors  $(\gamma(t + \frac{1}{n}))_{n \geq n_0}$  est une suite d'éléments qui ne sont pas dans  $P$ , et donc qui sont dans  $P'$  (car ils sont dans  $S(0, 1)$ )  
Comme  $\gamma$  est continue, elle converge vers  $\gamma(t)$ , et comme  $P'$  est fermé, on a donc  $\gamma(t) \in P'$   
Donc  $\gamma(t) \in P \cap P'$

On a donc que  $P \cap P'$  est non vide

Soit  $q$  un élément de  $P \cap P'$

On a  $q \in p_0(A)$  donc il existe  $\mu \in \mathbb{R}^+$  tel que  $\mu q \in A$

De même  $-q \in p_0(A)$  donc il existe  $\nu \in \mathbb{R}^+$  tel que  $-\nu q \in A$

$0_E \notin A$  donc  $\mu \neq 0$  et  $\nu \neq 0$

on note  $\alpha = \frac{\frac{1}{\nu}}{\frac{1}{\mu} + \frac{1}{\nu}}$  (on a  $\alpha \in [0, 1]$ )

Ainsi  $0_E = \alpha \cdot \mu q + (1 - \alpha) \cdot (-\nu q)$

Donc  $0_E \in \{c \cdot u + (1 - c) \cdot v, c \in [0, 1], (u, v) \in A^2\}$

Et donc  $0_E \in \text{conv}(A)$

□

## C Annexe